

# project2\_109550206

軟體定義網路及網路功能虛擬化 Lab2

ID: 109550206 Name:陳品劭

Self link (<https://hackmd.io/@pinchen/SDNFVproject2>).

## Part 1

1. How many OpenFlow headers with type “OFPT\_FLOW\_MOD” and command “OFPPC\_ADD” are there among all the packets?

Ans: 6

Wireshark抓到的狀況如下:

openflow_v5.type == 14						
No.	Time	Source	Destination	Protocol	Length	Info
40	0.225024138	127.0.0.1	127.0.0.1	OpenFlow	170	Type: OFPT_BARRIER_REQUEST
42	0.244355897	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
47	0.260642143	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
193	9.910860391	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
276	18.3099087...	127.0.0.1	127.0.0.1	OpenFlow	290	Type: OFPT_BARRIER_REQUEST
375	26.6415119...	127.0.0.1	127.0.0.1	OpenFlow	258	Type: OFPT_FLOW_MOD
376	26.6419826...	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
377	26.6420301...	127.0.0.1	127.0.0.1	OpenFlow	162	Type: OFPT_FLOW_MOD
378	26.6426712...	127.0.0.1	127.0.0.1	OpenFlow	170	Type: OFPT_FLOW_MOD
379	26.6428924...	127.0.0.1	127.0.0.1	OpenFlow	170	Type: OFPT_FLOW_MOD

共有以下12個openflow headers with type “OFPT\_FLOW\_MOD” and command “OFPPC\_ADD”

<div>OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 23 Cookie: 0x00010000ea6f4b8e Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPPC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 40000 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) .....0 = Has mask: False Length: 2 Value: ARP (0x0806) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCL_NO_BUFFER (65535) Pad: 000000000000</div>	<div>OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 25 Cookie: 0x000100007a585b6f Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPPC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 40000 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) .....0 = Has mask: False Length: 2 Value: Unknown (0x8942) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCL_NO_BUFFER (65535) Pad: 000000000000</div>	<div>OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 24 Cookie: 0x000100009465555a Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPPC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 40000 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) .....0 = Has mask: False Length: 2 Value: 802.1 Link Layer Discovery Protocol (LLDP) (0x88cc) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCL_NO_BUFFER (65535) Pad: 000000000000</div>
---	---	--

<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 26 Cookie: 0x00010000021b41dc Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 5 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) ... ..0 = Has mask: False Length: 2 Value: IPv4 (0x0000) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCLM_NO_BUFFER (65535) Pad: 000000000000 </pre>	<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 104 Transaction ID: 30 Cookie: 0x000900000b8d756f Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 10 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 32 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 000. = Field: OFPXM_OFB_IN_PORT (0) ... ..0 = Has mask: False Length: 4 Value: 2 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 011. = Field: OFPXM_OFB_ETH_DST (3) ... ..0 = Has mask: False Length: 6 Value: 26:3b:dc:f3:fa:1c (26:3b:dc:f3:fa:1c) OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 100. = Field: OFPXM_OFB_ETH_SRC (4) ... ..0 = Has mask: False Length: 6 Value: 76:f4:8d:0f:3f:86 (76:f4:8d:0f:3f:86) Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: 2 Max length: 0 Pad: 000000000000 </pre>	<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 104 Transaction ID: 29 Cookie: 0x000900000ba25d62f Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 10 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 32 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 000. = Field: OFPXM_OFB_IN_PORT (0) ... ..0 = Has mask: False Length: 4 Value: 1 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 011. = Field: OFPXM_OFB_ETH_DST (3) ... ..0 = Has mask: False Length: 6 Value: 76:f4:8d:0f:3f:86 (76:f4:8d:0f:3f:86) OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 100. = Field: OFPXM_OFB_ETH_SRC (4) ... ..0 = Has mask: False Length: 6 Value: 26:3b:dc:f3:fa:1c (26:3b:dc:f3:fa:1c) Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: 2 Max length: 0 Pad: 000000000000 </pre>
<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 0 Cookie: 0x00010000021b41dc Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 5 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) ... ..0 = Has mask: False Length: 2 Value: IPv4 (0x0000) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCLM_NO_BUFFER (65535) Pad: 000000000000 </pre>	<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 0 Cookie: 0x0001000007a585b6f Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 40000 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) ... ..0 = Has mask: False Length: 2 Value: Unknown (0x8942) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCLM_NO_BUFFER (65535) Pad: 000000000000 </pre>	<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 0 Cookie: 0x000100000465555a Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 40000 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) ... ..0 = Has mask: False Length: 2 Value: 802.1 Link Layer Discovery Protocol (LLDP) (0x88cc) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCLM_NO_BUFFER (65535) Pad: 000000000000 </pre>
<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 96 Transaction ID: 0 Cookie: 0x000100000ea6f4bde Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 40000 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 10 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 101. = Field: OFPXM_OFB_ETH_TYPE (5) ... ..0 = Has mask: False Length: 2 Value: ARP (0x0806) Pad: 000000000000 Instruction Type: OFPIT_CLEAR_ACTIONS (5) Length: 8 Pad: 00000000 Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: OFPP_CONTROLLER (4294967293) Max length: OFPCLM_NO_BUFFER (65535) Pad: 000000000000 </pre>	<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 104 Transaction ID: 0 Cookie: 0x000900000ba25d62f Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 10 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 32 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 000. = Field: OFPXM_OFB_IN_PORT (0) ... ..0 = Has mask: False Length: 4 Value: 1 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 011. = Field: OFPXM_OFB_ETH_DST (3) ... ..0 = Has mask: False Length: 6 Value: 76:f4:8d:0f:3f:86 (76:f4:8d:0f:3f:86) OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 100. = Field: OFPXM_OFB_ETH_SRC (4) ... ..0 = Has mask: False Length: 6 Value: 26:3b:dc:f3:fa:1c (26:3b:dc:f3:fa:1c) Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: 2 Max length: 0 Pad: 000000000000 </pre>	<pre> OpenFlow 1.4 Version: 1.4 (0x05) Type: OFPT_FLOW_MOD (14) Length: 104 Transaction ID: 0 Cookie: 0x000900000b8d756f Cookie mask: 0x0000000000000000 Table ID: 0 Command: OFPFC_ADD (0) Idle timeout: 0 Hard timeout: 0 Priority: 10 Buffer ID: OFP_NO_BUFFER (4294967295) Out port: OFPP_ANY (4294967295) Out group: OFPG_ANY (4294967295) Flags: 0x0001 Importance: 0 Match Type: OFPMT_OXM (1) Length: 32 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 000. = Field: OFPXM_OFB_IN_PORT (0) ... ..0 = Has mask: False Length: 4 Value: 2 OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 011. = Field: OFPXM_OFB_ETH_DST (3) ... ..0 = Has mask: False Length: 6 Value: 26:3b:dc:f3:fa:1c (26:3b:dc:f3:fa:1c) OXM field Class: OFPXM_OPENFLOW_BASIC (0x8000) 0000 100. = Field: OFPXM_OFB_ETH_SRC (4) ... ..0 = Has mask: False Length: 6 Value: 76:f4:8d:0f:3f:86 (76:f4:8d:0f:3f:86) Instruction Type: OFPIT_APPLY_ACTIONS (4) Length: 24 Pad: 00000000 Action Type: OFPAT_OUTPUT (0) Length: 16 Port: 2 Max length: 0 Pad: 000000000000 </pre>

據觀察共前6個分別與後6個相同。  
Value分別為

1. ARP (0x0806)
2. Unknown (0x8942) (BDDP)

3. 802.1 LLDP (0x88cc)
4. IPv4 (0x0800)
- 剩下兩個則是 IN\_PORT / ETH\_DST / ETH\_SRC (\*fwd)
5. 2 / 26:3b:dc:f3:fa:1c / 76:f4:8d:0f:3f:86
6. 1 / 76:f4:8d:0f:3f:86 / 26:3b:dc:f3:fa:1c

## 2. What are the match fields and the corresponding actions in each "OFPT\_FLOW\_MOD" message?

前四種的Actions是

1. CLEAR \* **Clears all actions from the datapath action set** \*
2. APPLY \* **Applies the action(s) immediately** \* (OUTPUT) \* **Output to switch port.** \*

後兩種則是只有

1. APPLY (OUTPUT)

## 3. What are the value of timeout for each flow rule installed in s1?

前四種flow rule 的time out 是0

後兩種的flow rule(fwd)的 time out 是 10



The screenshot displays a network switch's configuration interface. At the top, there is a blue icon of a switch and a large hexadecimal value: **0x8900006b8d756f**. Below this, a list of flow rule attributes is shown:

- Flow ID :** 0x8900006b8d756f
- State :** Added
- Bytes :** 0
- Packets :** 0
- Duration :** 4
- Flow Priority :** 10
- Table Name :** 0
- App Name :** \*fwd
- App ID :** 137
- Group ID :** 0x0
- Idle Timeout :** 10
- Hard Timeout :** 0
- Permanent :** false

---

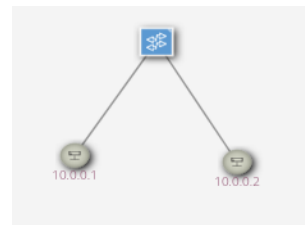
**Selector**

**ETH\_TYPE :** IN\_PORT:2,ETH\_DST:26:3B:DC:F3:FA:1C,ETH\_SRC:76:F4:8D:0F:3F:86

## Part 2 Install Flow Rules

於default環境下進行，如下:

```
demo@root > apps -a -s
* 6 org.onosproject.lldpprovider 2.2.0 LLDP Link Provider 16:06:08
* 15 org.onosproject.hostprovider 2.2.0 Host Location Provider
* 16 org.onosproject.optical-model 2.2.0 Optical Network Model
* 17 org.onosproject.openflow-base 2.2.0 OpenFlow Base Provider
* 18 org.onosproject.openflow 2.2.0 OpenFlow Provider Suite
* 55 org.onosproject.drivers 2.2.0 Default Drivers
* 140 org.onosproject.gui2 2.2.0 ONOS GUI2
demo@root >
```



第一次成功的json檔如下，一開始沒想太多就多給ETH\_TYPE(ARP)，然後確實可以arping通，但無法ping通。

```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "selector": {
    "criteria": [
      {
        "type": "IN_PORT",
        "port": "1"
      },
      {
        "type": "ETH_TYPE",
        "ethType": "0x0806"
      }
    ]
  },
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "2"
      }
    ]
  }
}
```

```
{
  "priority": 51000,
  "timeout": 0,
  "isPermanent": true,
  "selector": {
    "criteria": [
      {
        "type": "IN_PORT",
        "port": "2"
      },
      {
        "type": "ETH_TYPE",
        "ethType": "0x0806"
      }
    ]
  },
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "1"
      }
    ]
  }
}
```

後再新增兩個，IPV4\_DST的json (其default 要ETH\_TYPE(IPv4))，即可ping通，另外嘗試取消arping的flow rule，仍可ping通，但無法arping。應該是flow rule直接指定了port，所以沒用到mac。

```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "selector": {
    "criteria": [
      {
        "type": "IN_PORT",
        "port": "2"
      },
      {
        "type": "ETH_TYPE",
        "ethType": "0x0800"
      },
      {
        "type": "IPV4_DST",
        "ip": "10.0.0.1/32"
      }
    ]
  },
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "1"
      }
    ]
  }
}
```

```
{
  "priority": 49000,
  "timeout": 0,
  "isPermanent": true,
  "selector": {
    "criteria": [
      {
        "type": "IN_PORT",
        "port": "1"
      },
      {
        "type": "ETH_TYPE",
        "ethType": "0x0800"
      },
      {
        "type": "IPV4_DST",
        "ip": "10.0.0.2/32"
      }
    ]
  },
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "2"
      }
    ]
  }
}
```

## 新增flow rule command

```
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @flows_s1-1_109550206.json 'http://localhost:8181/onos/v1/flows/of:000000000000001'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @flows_s1-2_109550206.json 'http://localhost:8181/onos/v1/flows/of:000000000000001'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @flows_s1-3_109550206.json 'http://localhost:8181/onos/v1/flows/of:000000000000001'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @flows_s1-4_109550206.json 'http://localhost:8181/onos/v1/flows/of:000000000000001'
demo@SDN-NFV:~/Desktop$
```

## arping、ping成功 command

```
mininet> h1 arping h2
ARPING 10.0.0.2 from 10.0.0.1 h1-eth0
Unicast reply from 10.0.0.2 [32:4F:49:2B:62:C1] 0.758ms
Unicast reply from 10.0.0.2 [32:4F:49:2B:62:C1] 0.718ms
Unicast reply from 10.0.0.2 [32:4F:49:2B:62:C1] 0.579ms
Unicast reply from 10.0.0.2 [32:4F:49:2B:62:C1] 0.533ms
Unicast reply from 10.0.0.2 [32:4F:49:2B:62:C1] 0.576ms
^C Sent 5 probes (1 broadcast(s))
Received 5 response(s)
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.295 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.188 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4109ms
rtt min/avg/max/mdev = 0.032/0.117/0.295/0.107 ms
mininet>
```

後來於part4時思考，是否我的flow rule不太對，嘗試讓flow rule跟fwd的差不多:

註:下列flow rule 皆為雙向都下且mac是有給的(??只是示意)

```
1  flow_rule_A:
2  {
3      "priority": 50000,
4      "timeout": 0,
5      "isPermanent": true,
6      "selector": {
7          "criteria": [
8              {
9                  "type": "IN_PORT",
10                 "port": "1"
11             },
12             {
13                 "type": "ETH_DST",
14                 "mac": "?:?:?:?:?:?:?:?"
15             },
16             {
17                 "type": "ETH_SRC",
18                 "mac": "?:?:?:?:?:?:?:?"
19             },
20         ]
21     },
22     "treatment": {
23         "instructions": [
24             {
25                 "type": "OUTPUT",
26                 "port": "2"
27             }
28         ]
29     }
30 }
```

照著fwd的flow rule去下，如flow\_rule\_A，發現arping不通。

```

1  flow_rule_B:
2  {
3      "priority": 50000,
4      "timeout": 0,
5      "isPermanent": true,
6      "selector": {
7          "criteria": [
8              {
9                  "type": "IN_PORT",
10                 "port": "1"
11             },
12             {
13                 "type": "ETH_TYPE",
14                 "ethType": "0x0806"
15             },
16             {
17                 "type": "ETH_DST",
18                 "mac": "?:?:?:?:?:?:?:?"
19             },
20             {
21                 "type": "ETH_SRC",
22                 "mac": "?:?:?:?:?:?:?:?"
23             },
24         ]
25     },
26     "treatment": {
27         "instructions": [
28             {
29                 "type": "OUTPUT",
30                 "port": "2"
31             }
32         ]
33     }
34 }

```

把ETH\_TYPE拿回來，如flow\_rule\_B，還是arping不通。

```
1  flow_rule_C:
2  {
3      "priority": 50000,
4      "timeout": 0,
5      "isPermanent": true,
6      "selector": {
7          "criteria": [
8              {
9                  "type": "IN_PORT",
10                 "port": "1"
11             },
12             {
13                 "type": "ETH_TYPE",
14                 "ethType": "0x0806"
15             },
16             {
17                 "type": "ETH_SRC",
18                 "mac": "?:?:?:?:?:?:?:?:?"
19             },
20         ]
21     },
22     "treatment": {
23         "instructions": [
24             {
25                 "type": "OUTPUT",
26                 "port": "2"
27             }
28         ]
29     }
30 }
```

把ETH\_DST拿掉，如flow\_rule\_C，arping通了。



```

1  flow_rule_D:
2  {
3      "priority": 50000,
4      "timeout": 0,
5      "isPermanent": true,
6      "selector": {
7          "criteria": [
8              {
9                  "type": "ETH_TYPE",
10                 "ethType": "0x0806"
11             },
12             {
13                 "type": "ETH_SRC",
14                 "mac": "?:?:?:?:?:?:?:?"
15             },
16         ]
17     },
18     "treatment": {
19         "instructions": [
20             {
21                 "type": "OUTPUT",
22                 "port": "2"
23             }
24         ]
25     }
26 }

```

然後覺得IN\_PORT應該也不用，如flow\_rule\_D，arping也通了。

```

1  flow_rule_E:
2  {
3      "priority": 50000,
4      "timeout": 0,
5      "isPermanent": true,
6      "selector": {
7          "criteria": [
8              {
9                  "type": "ETH_SRC",
10                 "mac": "?:?:?:?:?:?:?:?"
11             }
12         ]
13     },
14     "treatment": {
15         "instructions": [
16             {
17                 "type": "OUTPUT",
18                 "port": "2"
19             }
20         ]
21     }
22 }

```

後來覺得或許ETH\_TYPE也不用，如flow\_rule\_E，不過失敗了，arping不通。

結論1:ETTH\_TYPE一定要下。

結論2:IN\_PORT or ETH\_SRC 可以任取(至少一個)都有的話即指定這個port且是這個mac的才行。

接著仍然覺得很奇怪，為什麼不能下ETH\_DST，偶然同時下了flow\_rule\_B、flow\_rule\_C，發現以下狀況，第一封broadcast是透過flow\_rule\_C，但接下來皆是flow\_rule\_B，另外還發現，如果flow\_rule\_C只下單向，h2 arping h1不會通。

Added	1	40	50000	0	IN_PORT:1, ETH_SRC:3A:78:D5:7 A:E5:7D, ETH_TYPE:arp	imm[OUTPUT:2], cleared:false	*rest
Added	4	66	50000	0	IN_PORT:1, ETH_DST:BA:6D:DC:C 4:3D:59, ETH_SRC:3A:78:D5:7 A:E5:7D, ETH_TYPE:arp	imm[OUTPUT:2], cleared:false	*rest
Added	5	64	50000	0	IN_PORT:2, ETH_DST:3A:78:D5:7 A:E5:7D, ETH_SRC:BA:6D:DC:C 4:3D:59, ETH_TYPE:arp	imm[OUTPUT:1], cleared:false	*rest

結論3:下ETH\_DST的話，不會處理broadcast的封包所以無法通。

另外也嘗試了只多開ETH\_TYPE:IPv4/OUTPUT:CONTROLLER，能否ping通，結果無法，fwd背後有做其他事情來彌補flow rule的問題。

## Part 3 Create Topology with Broadcast Storm

python script and GUI

```
from mininet.topo import Topo
class Project2_Topo_109550206( Topo ):
    def __init__(self):
        Topo.__init__(self)
        #Add three switches
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        #Add two hosts
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        #Add link H with S
        self.addLink(h1, s1)
        self.addLink(h2, s2)
        #Add links to make loop
        self.addLink(s1, s2)
        self.addLink(s2, s3)
        self.addLink(s3, s1)
topos = { 'topo': Project2_Topo_109550206 }
```

**ONOS Summary**

Version :	2.2.0
Devices :	3
Links :	2
Hosts :	0
Topology SCCs :	3
Intents :	0
Flows :	15

**Infrastructure Link**

A type :	Device
A id :	of:0000000000000001
A name :	of:0000000000000001
A port :	3
B type :	Device
B id :	of:0000000000000003
B name :	of:0000000000000003
B port :	2
A to B :	[no link]
B to A :	direct / active / not expected

根據GUI顯示的port去下flow rule使其傳輸路徑有loop，即會Broadcast Storm。此處flow rule就只給IN\_PORT/OUTPUT×N，如下：

```
{
  "priority": 50000,
  "timeout": 0,
  "isPermanent": true,
  "selector": {
    "criteria": [
      {
        "type": "IN_PORT",
        "port": "1"
      }
    ]
  },
  "treatment": {
    "instructions": [
      {
        "type": "OUTPUT",
        "port": "2"
      },
      {
        "type": "OUTPUT",
        "port": "3"
      }
    ]
  }
}
```

```
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @1_1.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @1_2.json 'http://localhost:8181/onos/v1/flows/of:0000000000000001'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @2_1.json 'http://localhost:8181/onos/v1/flows/of:0000000000000002'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @2_2.json 'http://localhost:8181/onos/v1/flows/of:0000000000000002'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @3_1.json 'http://localhost:8181/onos/v1/flows/of:0000000000000003'
demo@SDN-NFV:~/Desktop$ curl -u onos:rocks -X POST -H 'Content-Type:application/json' -d @3_1.json 'http://localhost:8181/onos/v1/flows/of:0000000000000003'
demo@SDN-NFV:~/Desktop$
```

我設計的路線為:

h1 → s1  
h2 → s2  
s1 → h1  
s1 → s2  
s1 → s3  
s2 → h2  
s2 → s1  
s3 → s2

When h1 ping h2

封包路線:

h1 → s1 → s2 → h2  
s1 → s3 → s2 → h2  
s1 → s3 → s2 → s1 → s3 (loop)

When h2 ping h1

封包路線:

h2 → s2 → s1 → h1  
s1 → s3 → s2 → s1 (loop)

名稱	狀態	38% CPU	87% 記憶體	1% 磁碟	0% 網路
> Google Chrome (21)		0.3%	1,323.0 ...	0.1 MB/秒	0 Mbps
Dropbox (32 位元)		0.1%	137.2 MB	0 MB/秒	0 Mbps
> VirtualBox Virtual Machine		31.5%	82.8 MB	0.1 MB/秒	0 Mbps
> Trend Micro Anti-Malware Sol...		0.1%	75.0 MB	0.1 MB/秒	0 Mbps

註:(DUP!)產生速度快過顯示

## Part 4 Trace ReactiveForwarding

下兩圖皆為已開啟fwd後，再開啟wireshark後，h1 ping h2 -c 1之後，停止wireshark的封包

1	0.000000000	fe80::30f5:70ff:fed...	ff02::fb	MDNS	203	Standard query response 0x0000 PTR, cache...
2	0.104920726	127.0.0.1	127.0.0.1	OpenFlow	426	Type: OFPT_PACKET_OUT
3	0.104976094	127.0.0.1	127.0.0.1	OpenFlow	426	Type: OFPT_PACKET_OUT
4	0.105137945	02:eb:d0:78:9a:04		LLDP	141	TTL = 120
5	0.105174675	02:eb:d0:78:9a:04		0x8942	141	Sent by us
6	0.105197216	02:eb:d0:78:9a:04		LLDP	141	TTL = 120
7	0.105211602	127.0.0.1	127.0.0.1	TCP	68	34986 → 6653 [ACK] Seq=1 Ack=717 Win=86 L...
8	0.105232803	02:eb:d0:78:9a:04		0x8942	141	Sent by us
9	0.326042824	ce:08:38:b8:47:a0		ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
10	0.326242107	127.0.0.1	127.0.0.1	OpenFlow	152	Type: OFPT_PACKET_IN
11	0.326254149	127.0.0.1	127.0.0.1	TCP	68	6653 → 34986 [ACK] Seq=717 Ack=85 Win=86 ...
12	0.326734560	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_OUT
13	0.326827897	ce:08:38:b8:47:a0		ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
14	0.326837472	c6:bd:d7:8d:4c:63		ARP	44	10.0.0.2 is at c6:bd:d7:8d:4c:63
15	0.326970823	127.0.0.1	127.0.0.1	OpenFlow	152	Type: OFPT_PACKET_IN
16	0.328110401	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_OUT
17	0.328178839	c6:bd:d7:8d:4c:63		ARP	44	10.0.0.2 is at c6:bd:d7:8d:4c:63
18	0.328184766	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x4091, seq=1/256...
19	0.328285383	127.0.0.1	127.0.0.1	OpenFlow	208	Type: OFPT_PACKET_IN
20	0.328731024	127.0.0.1	127.0.0.1	TCP	144	8181 → 55250 [PSH, ACK] Seq=1 Ack=1 Win=8...
21	0.328739645	127.0.0.1	127.0.0.1	TCP	68	55250 → 8181 [ACK] Seq=1 Ack=77 Win=86 Le...
22	0.328800280	127.0.0.1	127.0.0.1	TCP	95	8181 → 55250 [PSH, ACK] Seq=77 Ack=1 Win=...
23	0.328828498	127.0.0.1	127.0.0.1	TCP	68	55250 → 8181 [ACK] Seq=1 Ack=104 Win=86 L...
24	0.329285319	127.0.0.1	127.0.0.1	OpenFlow	206	Type: OFPT_PACKET_OUT
25	0.329419714	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x4091, seq=1/256...
26	0.329431556	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x4091, seq=1/256...
27	0.329546390	127.0.0.1	127.0.0.1	OpenFlow	208	Type: OFPT_PACKET_IN
28	0.330063398	127.0.0.1	127.0.0.1	OpenFlow	206	Type: OFPT_PACKET_OUT
29	0.330819279	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x4091, seq=1/256...
30	0.331078236	127.0.0.1	127.0.0.1	OpenFlow	180	Type: OFPT_BARRIER_REQUEST
31	0.331136138	127.0.0.1	127.0.0.1	TCP	68	34986 → 6653 [ACK] Seq=449 Ack=1269 Win=8...
32	0.331144039	127.0.0.1	127.0.0.1	OpenFlow	76	Type: OFPT_BARRIER_REPLY
33	0.334080636	127.0.0.1	127.0.0.1	TCP	103	8181 → 55250 [PSH, ACK] Seq=104 Ack=1 Win...
34	0.334083356	127.0.0.1	127.0.0.1	TCP	68	55250 → 8181 [ACK] Seq=1 Ack=139 Win=86 L...
35	0.334133071	127.0.0.1	127.0.0.1	TCP	94	8181 → 55250 [PSH, ACK] Seq=139 Ack=1 Win...
36	0.334135097	127.0.0.1	127.0.0.1	TCP	68	55250 → 8181 [ACK] Seq=1 Ack=165 Win=86 L...
37	0.357332801	fe80::c4bd:d7ff:fe8...	ff02::2	ICMPv6	72	Router Solicitation from c6:bd:d7:8d:4c:63

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	79	55794 → 8181 [PSH, ACK] Seq=1 Ack=1 Win=8...
2	0.014672266	127.0.0.1	127.0.0.1	TCP	95	8181 → 55794 [PSH, ACK] Seq=1 Ack=12 Win=...
3	0.014700457	127.0.0.1	127.0.0.1	TCP	68	55794 → 8181 [ACK] Seq=12 Ack=28 Win=86 L...
4	0.339417599	e2:14:ec:56:73:9a	127.0.0.1	ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
5	0.339736346	127.0.0.1	127.0.0.1	OpenFlow	152	Type: OFPT_PACKET_IN
6	0.339748328	127.0.0.1	127.0.0.1	TCP	68	6653 → 35054 [ACK] Seq=1 Ack=85 Win=86 Le...
7	0.340472185	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_OUT
8	0.340474955	127.0.0.1	127.0.0.1	TCP	68	35054 → 6653 [ACK] Seq=85 Ack=83 Win=86 L...
9	0.343117720	e2:14:ec:56:73:9a	127.0.0.1	ARP	44	Who has 10.0.0.2? Tell 10.0.0.1
10	0.343132587	e6:4f:7a:05:ed:9e	127.0.0.1	ARP	44	10.0.0.2 is at e6:4f:7a:05:ed:9e
11	0.343265035	127.0.0.1	127.0.0.1	OpenFlow	152	Type: OFPT_PACKET_IN
12	0.345861881	127.0.0.1	127.0.0.1	TCP	145	8181 → 55794 [PSH, ACK] Seq=28 Ack=12 Win...
13	0.345870783	127.0.0.1	127.0.0.1	TCP	68	55794 → 8181 [ACK] Seq=12 Ack=105 Win=86 ...
14	0.345916488	127.0.0.1	127.0.0.1	TCP	95	8181 → 55794 [PSH, ACK] Seq=105 Ack=12 Wi...
15	0.345919973	127.0.0.1	127.0.0.1	TCP	68	55794 → 8181 [ACK] Seq=12 Ack=132 Win=86 ...
16	0.345955592	127.0.0.1	127.0.0.1	OpenFlow	150	Type: OFPT_PACKET_OUT
17	0.346028456	e6:4f:7a:05:ed:9e	127.0.0.1	ARP	44	10.0.0.2 is at e6:4f:7a:05:ed:9e
18	0.346034979	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x41a2, seq=1/256...
19	0.346152165	127.0.0.1	127.0.0.1	OpenFlow	208	Type: OFPT_PACKET_IN
20	0.349159697	127.0.0.1	127.0.0.1	OpenFlow	206	Type: OFPT_PACKET_OUT
21	0.349300738	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x41a2, seq=1/256...
22	0.349324079	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x41a2, seq=1/256...
23	0.349455080	127.0.0.1	127.0.0.1	OpenFlow	208	Type: OFPT_PACKET_IN
24	0.350367987	127.0.0.1	127.0.0.1	OpenFlow	206	Type: OFPT_PACKET_OUT
25	0.350426053	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x41a2, seq=1/256...
26	0.350874966	127.0.0.1	127.0.0.1	OpenFlow	180	Type: OFPT_BARRIER_REQUEST
27	0.350918266	127.0.0.1	127.0.0.1	TCP	68	35054 → 6653 [ACK] Seq=449 Ack=553 Win=86...
28	0.350924879	127.0.0.1	127.0.0.1	OpenFlow	76	Type: OFPT_BARRIER_REPLY
29	0.355435010	127.0.0.1	127.0.0.1	TCP	103	8181 → 55794 [PSH, ACK] Seq=132 Ack=12 Wi...
30	0.355440622	127.0.0.1	127.0.0.1	TCP	68	55794 → 8181 [ACK] Seq=12 Ack=167 Win=86 ...
31	0.355525645	127.0.0.1	127.0.0.1	TCP	94	8181 → 55794 [PSH, ACK] Seq=167 Ack=12 Wi...
32	0.355527867	127.0.0.1	127.0.0.1	TCP	68	55794 → 8181 [ACK] Seq=12 Ack=193 Win=86 ...
33	0.393756805	127.0.0.1	127.0.0.1	TCP	68	6653 → 35054 [ACK] Seq=553 Ack=457 Win=86...
34	1.363504049	127.0.0.1	127.0.0.1	OpenFlow	140	Type: OFPT_MULTIPART_REQUEST
35	1.363718462	127.0.0.1	127.0.0.1	OpenFlow	572	Type: OFPT_MULTIPART_REPLY
36	1.363734591	127.0.0.1	127.0.0.1	TCP	68	6653 → 35054 [ACK] Seq=625 Ack=961 Win=86...
37	1.364219683	127.0.0.1	127.0.0.1	OpenFlow	6180	Type: OFPT_MULTIPART_REPLY
38	1.364233645	127.0.0.1	127.0.0.1	TCP	68	6653 → 35054 [ACK] Seq=625 Ack=7073 Win=7...
39	1.386059284	127.0.0.1	127.0.0.1	OpenFlow	92	Type: OFPT_MULTIPART_REQUEST
40	1.386384449	127.0.0.1	127.0.0.1	OpenFlow	996	Type: OFPT_MULTIPART_REPLY[Malformed Pack...
41	1.430453493	127.0.0.1	127.0.0.1	TCP	68	6653 → 35054 [ACK] Seq=649 Ack=8001 Win=8...
42	1.552252572	127.0.0.1	127.0.0.1	OpenFlow	132	Type: OFPT_MULTIPART_REQUEST
43	1.552476730	127.0.0.1	127.0.0.1	OpenFlow	84	Type: OFPT_MULTIPART_REPLY
44	1.552509991	127.0.0.1	127.0.0.1	OpenFlow	84	Type: OFPT_MULTIPART_REPLY
45	1.552531419	127.0.0.1	127.0.0.1	OpenFlow	84	Type: OFPT_MULTIPART_REPLY
46	1.552927898	127.0.0.1	127.0.0.1	TCP	68	6653 → 35054 [ACK] Seq=713 Ack=8049 Win=8...

Frame 4: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface 0

Linux cooked capture

Packet type: Broadcast (1)

Link-layer address type: 1

Link-layer address length: 6

Source: e2:14:ec:56:73:9a (e2:14:ec:56:73:9a)

Unused: 0000

Protocol: ARP (0x0806)

Address Resolution Protocol (request)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

Sender MAC address: e2:14:ec:56:73:9a (e2:14:ec:56:73:9a)

Sender IP address: 10.0.0.1

Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)

Target IP address: 10.0.0.2

Packet type (sll.pktttype), 2 bytes

Packets: 46 · Displayed: 46 (100.0%) · Dropped: 0 (0.0%) Profile: Classic

1. h1丟出一個ARP (Who has 10.0.0.2? Tell 10.0.0.1)
2. s1收到後miss (Flow table沒有對應的方法，s1 does not know who is 10.0.0.2)
3. s1丟一個Open Flow去問c1 (OFPT\_PACKET\_IN: IN\_PORT:1)
4. c1回s1 (OFPT\_PACKET\_OUT: OUTPUT:2)
5. s1把該ARP從Port 2轉給h2
6. h2收到ARP，因h2即為10.0.0.2，因此回一個ARP (10.0.0.2 at ????:????:????:???)
7. s1收到後，仍然miss (還沒有對應的flow rule下來)
8. s1丟一個Open Flow去問c1 (OFPT\_PACKET\_IN: IN\_PORT:2)
9. c1回s1 (OFPT\_PACKET\_OUT: OUTPUT:1)
10. s1把該ARP從Port 1轉給h1
11. h1收到，得知10.0.0.2在哪個mac，並丟ICMP過去

12. s1收到後，仍然miss (還沒有對應的flow rule下來)
13. s1丟一個Open Flow去問c1 (OFPT\_PACKET\_IN: IN\_PORT:1)
14. c1回s1 (OFPT\_PACKET\_OUT: OUTPUT:2)
15. s1把該ICMP從Port 2轉給h2
16. h2收到first ICMP request

## What you've learned or solved

---

這次Lab學的東西有點多，實際去抓封包、研究、思考fwd在幹嘛，具體而言還是有一些模稜兩可不確定的地方，但整體而言多了解很多東西，也與老師上課講的架構、概念、方法有對到。一開始做part2還滿卡的，不知道怎麼下手，先隨便猜，然後過了，接著做part3時，覺得滿有趣的(可能是看到跑超快DUP和塊當掉的VM覺得有趣?!)，然後開始思考part4時，回去想part2的東西，這塊覺得是最有成就感的地方，相對於一開始更完整了解flow rule各個match field/instruction/action的功能、意義。